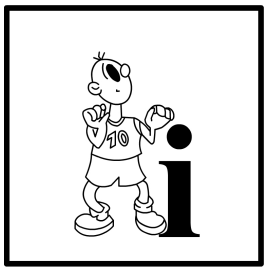
Idee, Zweck, Systembeschreibung:



Für Oldtimer-Rallyes brauchen *Personen* einen Schnittrechner. Hier wird oft mit Gleichmässigkeitsfahrt[1] gefahren. Das heißt, man muss über z.B. 5,4km hinweg einen gleichmässigen Schnitt von 35km/h fahren und die Auswerter stehen dann im Wald und schauen, ob wir zur richtigen *Zeit* vorbeikommen. Während der Ralley brauchen wir eine Anzeige, welche *Strecke* (in m auf zwei Kommastellen) zurücklegt werden muss. Mit dieser Information können wir den Tachometer vergleichen und in Echtzeit prüfen ob wir zu schnell oder zu langsam sind. Nach Eingabe des geforderten Schnitts *(Geschwindigkeit)*, soll auf der *Benutzeroberfläche* der *Sekundentakt (Zeit seit Start)*, angezeigt werden. Dieser soll durch eine *Schaltfläche (Startknopf)*, wie eine Stoppuhr gestartet werden können. Nach dem Bedienen des *Startknopfes* soll die *Soll-Strecke* in Echtzeit *berechnet* und *angezeigt* werden. Eine weitere Schaltfläche *(Zwischenzeit anzeigen)* soll ermöglichen, dass eine *Zwischenzeit* auf der Benutzeroberfläche *angezeigt* wird, wenn man diese Schaltfläche *bedient*. Gleichzeitig wird dann die bis dahin *gefahrene Strecke (Abschnitts-Strecke) angezeigt*. Mit einer *Reset-Schaltfläche* kann man den *Sekundentakt (Stoppuhr)* auf 00.00 zurücksetzen.

Anwendungsfälle:

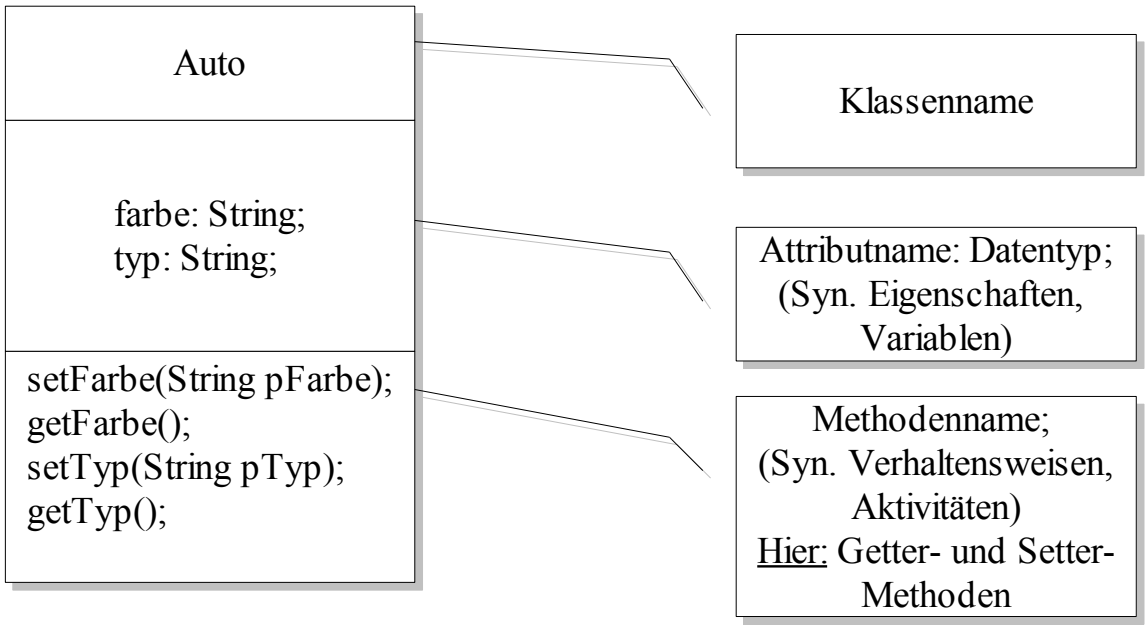
		
Myra Bellamy ist *Minifahrerin*. Sie muss einen *geforderten Schnitt* von *35 km/h (Geschwindigkeit)* fahren.	*Moe Szyslak i*st *Opelfahrer.* Er muss einen *geforderten Schnitt* von *20 km/h (Geschwindigkeit)* fahren.	*Marv Albert* ist *Mercedesfahrer.*. Er muss einen *geforderten Schnitt* von 50 *km/h (Geschwindigkeit)* fahren.

1 Die Gleichmäßigkeitsfahrt ist eine Disziplin bei heutigen Oldtimer-Rallyes und hat ihren Ursprung in den Wurzeln des Rallyesports der 1920er und 30er Jahre als zwischen den Zeitkontrollen durch den Veranstalter festgelegte Durchschnittsgeschwindigkeiten zu fahren waren. […] Die Schnitte dürfen allerdings nicht höher als 50 km/h liegen. Die Überwachung erfolgt entweder durch bekannte Zeitnahmen oder "Geheime Zeitkontrollen". Abweichungen von der Sollzeit werden mit Fehlerpunkten bestraft. Quelle: Wikimedia Inc.

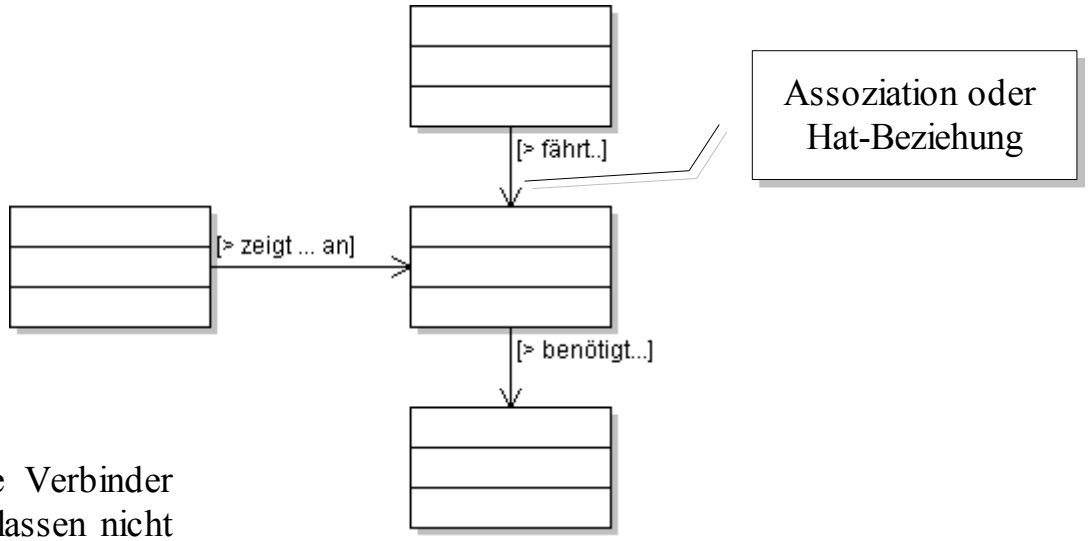
[Grafiken: The Simpsons TM, erstellt von CJ]

Merke: Sogenannte Getter- und Setter-Methoden[2] (Syn. Aktivität, Verhaltensweise) existieren für jede Eigenschaft. Z.B. Die Methoden setFarbe(String pFarbe) und getFarbe() für das Attribut (syn. Variable, Eigenschaft) String farbe der Klasse Auto. Diese Methoden dienen dazu Eigenschaftswerte einzelner Objekte zu modifizieren (bearbeiten, ändern) bzw. erstmalig zu initialisieren (Wert setzen). Es sind quasi Teilhandlungen auf unterster Ebene (Hinweis: kleinschrittig denken).

Zur Erinnerung die UML-Notation einer Klasse:



Vorschlag für die System-Architektur des Oldtimer-Ralley-Streckenplaner:



Hinweis: Ohne diese Verbinder (Pfeile) können die Klassen nicht miteinander kommunizieren.

2 „get“ steht für „holen“, „set“ steht für „setzen“

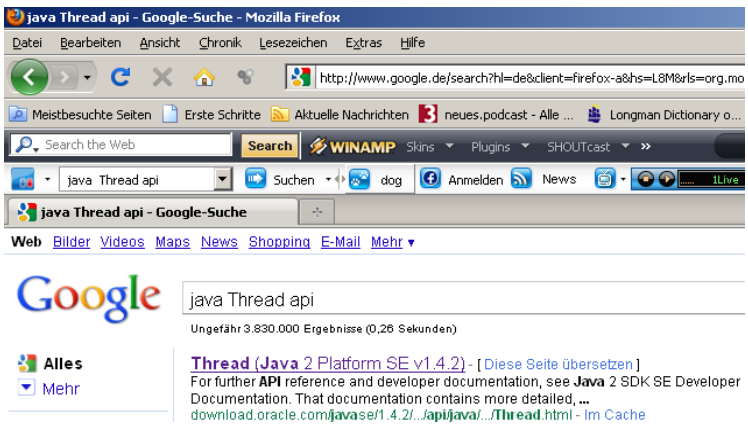
[Grafiken: The Simpsons TM, erstellt von CJ]

Beispiel Soll-Strecken-Formel (es gibt auch andere Lösungen):

sollstrecke = (sekundentakt * geschwindigkeit * 100)/4.166666666

Die bestehende Excel-Anwendung (stoppuhr.xls) soll in ähnlicher Form (siehe Anforderungen) durch eine Internetanwendung (WebApp) in Java ersetzt werden:

Die API



Die genutzten Methoden der Klasse Stoppuhr stammen teilweise aus der Klasse Thread. Diese ist Bestandteil der Standard-Bibliothek `java.lang.*` und wird API[3] (Applikation Programming Interface) genannt. „Googeln“ Sie nach den Stichworten *„java Thread api“*, dann können Sie auf die von *Oracle* online zur Verfügung stehende API zugreifen. Allgemein finden Sie auf diese Weise *„java <Klassenname> api“* immer einen Verweis auf die entsprechende Klasse in der zugehörigen API.

3 http://download.oracle.com/javase/1.4.2/docs/api/

[Grafiken: The Simpsons TM, erstellt von CJ]

Behandlung von Eingaben und Ausgaben der Hauptfenster-Klasse (GUI): Konvertierung in andere Datentypen

Heißt umgangssprachlich „umwandeln“. Da die Werte, die über die Benutzoberfläche eingegeben werden vom Datentyp String sind müssen Sie umgewandelt werden, um damit Rechnen zu können. Umgekehrt können auf der Benutzeroberfläche nur Strings angezeigt werden.

Man kann einen String (z.B. eine Eingabe über die Benutzeroberfläche) in einen Double umwandeln oder umgekehrt.

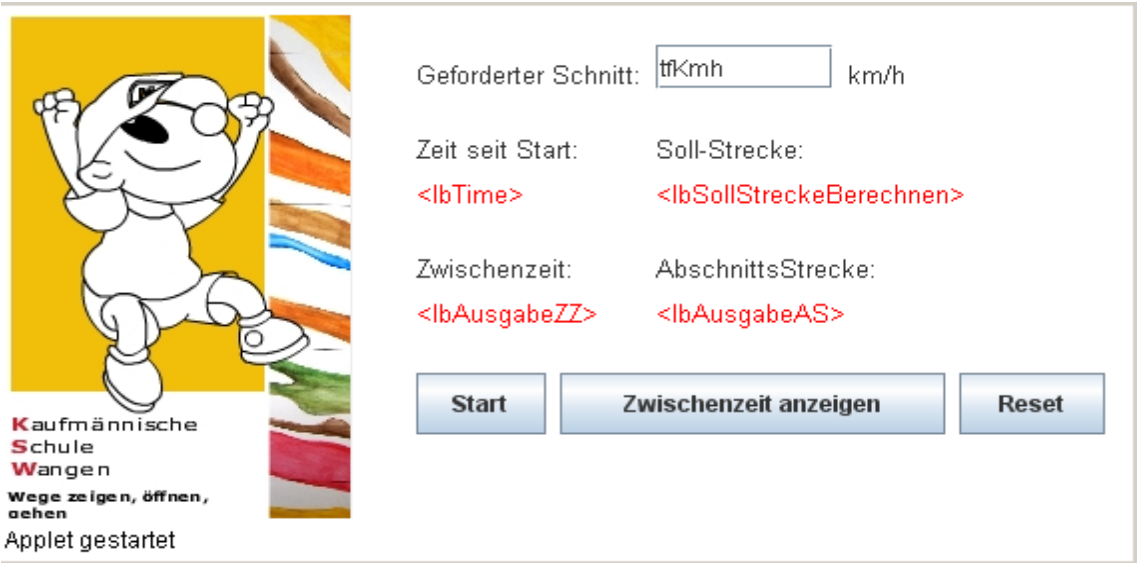
Beispiel für die Hauptfensterklasse-Klasse (GUI):

```
//Assoziation (Objekt)
Person person = new Person();

/*Eingabe:
 *Beispiel GUI-Klasse -  String in Double: Wandelt den String aus dem
 *Textfeld tfGewicht in den Datentyp double und setzt den Wert
 *in der Klasse Person*/

public void stringInDouble(){
    person.setGewicht(Double.valueOf(tfGewicht.getText()));
}

/*Ausgabe:
 *Beispiel GUI-Klasse - Double in String: Wandelt das Attribut double
 *gewicht in einen String und zeigt diesen im Textfeld
 *tfGewicht auf der Benutzeroberflaeche an*/

public void doubleInString(){
    tfgewicht.setText(String.valueOf(person.getGewicht()));
}
```

Die Klasse `Stoppuhr.java`

```
public class Stoppuhr extends Thread{

  // Anfang Attribute
  private boolean Stoppuhr;
  private long startZeit;
  private Hauptfenster gui;
  private double zeit;
  private double zeitInSek;
  // Ende Attribute

  public Stoppuhr(Hauptfenster gui) {
    this.gui = gui;
  }

  // Anfang Methoden
  public double getZeitInSek() {
    return zeitInSek;
  }

  public void setZeitInSek(double zeitInSek) {
    this.zeitInSek = zeitInSek;
  }

  public double getZeit() {
    return zeit;
  }

  public void setZeit(double zeit) {
    this.zeit = zeit;
  }

  public void berechneZeitInSek(double zeit){
    double zeitInSek = (double)(zeit/1000L);
    this.setZeitInSek(zeitInSek);
  }

  public void berechneZeitInStunden(double zeitInSek){
    double zeitInMin = (double)(zeitInSek/60);
    double zeitInStd = (double)(zeitInMin/60);
    this.setZeit(zeitInStd);
  }
  public void anhalten() {
        Stoppuhr = false;
  }


 public void run() {
  Stoppuhr=true;
  startZeit = System.currentTimeMillis();
  while(Stoppuhr == true){
   try{
    Thread.sleep(499);
   }  catch(InterruptedException ie) {

   }

    gui.aktualisiere(System.currentTimeMillis()-startZeit);
  }
 }
 // Ende Methoden

}
```

So könnte die grafische Benutzeroberfläche aussehen:



[Grafiken: The Simpsons TM, erstellt von CJ]